PR #40743 完整报告

vllm-project/vllm

[Test] Fix test_dynamic_shapes_compilation for torch 2.12

合并时间：2026-04-28 08:51

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40743

执行摘要

- 一句话：修复动态形状编译测试在 torch 2.12 中的脆弱性
- 推荐动作：该 PR 适合快速合入，解决 torch 2.12 升级后的测试回归。变更简单，review 已通过。值得关注的是「用更可靠的数值比较替代字符串断言」的测试设计思路，适用于其他脆弱测试场景。

功能与动机

在 torch 2.12 中，test_dynamic_shapes_compilation 因动态形状图回退到 *eager* 模式而断言 'no' == 'yes'，导致 12 个测试用例失败。原测试使用 GPT-2 评估自身输出是否为“垃圾”并断言结果是否为 "yes"，非常脆弱（不同 GPU 结果不同）。参考 Issue #180913。

实现拆解

1. 替换断言策略：移除原测试中基于 tokenizer 的 "yes"/"no" 二级生成和断言逻辑，改用 *check_logprobs_close* 比较 *compiled* 模型与 *eager* 模型的 logprobs 输出。
2. 新增 *eager* 模型参考：测试中先运行 *compiled* 模型生成输出，再以 *enforce_eager=True* 初始化同一模型作为参考，确保比较基准一致。
3. 增加 prompt 多样性：从单个 prompt 扩展到两个 prompt（"Hello, my name is" 和 "The *capital* of France is"），且每个 prompt 生成 5 个 token 并记录 logprobs，提升测试覆盖面。
4. 移除不需要的依赖：删除 *get_tokenizer* 的导入和相关 tokenizer 调用。
5. 导入 *check_logprobs_close*：从 tests.models.utils 添加该辅助函数的导入，用于 logprobs 数值比较。

关键文件：

- tests/compile/test_dynamic_shapes_compilation.py（模块 编译测试；类别 test；类型 test-coverage）：唯一变更文件，重写测试断言策略，从字符串断言改为 logprobs 数值比较。

关键符号：未识别

关键源码片段

tests/compile/test_dynamic_shapes_compilation.py

唯一变更文件，重写测试断言策略，从字符串断言改为 logprobs 数值比较。

```
# tests/compile/test_dynamic_shapes_compilation.py ( 关键变更部分 )

def test_dynamic_shapes_compilation(
    monkeypatch, model_name, shapes_type, use_aot_compile,
    use_bytecode_hook, evaluate_guards
):
    """Test that all dynamic shapes types compile successfully"""
    # ... 前序 setup 与旧代码相同 ...

    prompt = "Hello, my name is"

    # 初始化 compiled 模型 （与旧代码相同）
    model = LLM(
        model=model_name,
        compilation_config={
            "mode": CompilationMode.VLLM_COMPILE,
            "dynamic_shapes_config": {
                "type": shapes_type.value,
                "evaluate_guards": evaluate_guards,
            },
        },
        max_model_len=1024, # 显式指定以防默认值过大
    )

    # 使用 logprobs 采样参数替代原来的 "yes"/"no" 过滤
    sampling_params = SamplingParams(max_tokens=5, temperature=0, logprobs=10)
    test_prompts = [prompt, "The capital of France is"]

    # 收集 compiled 模型输出
    compiled_outputs = []
    for p in test_prompts:
        output = model.generate(p, sampling_params)[0].outputs[0]
        assert len(output.text.strip()) > 0, "Compiled model produced empty output"
        compiled_outputs.append((output.token_ids, output.text, output.logprobs))

    del model
    gc.collect()
    torch.accelerator.empty_cache()
    torch.accelerator.synchronize()

    # 用 eager 模式初始化参考模型
    eager_model = LLM(model=model_name, enforce_eager=True, max_model_len=1024)
    eager_outputs = []
    for p in test_prompts:
        output = eager_model.generate(p, sampling_params)[0].outputs[0]
        assert len(output.text.strip()) > 0, "Eager model produced empty output"
        eager_outputs.append((output.token_ids, output.text, output.logprobs))
```

```
del eager_model
gc.collect()
torch.accelerator.empty_cache()
torch.accelerator.synchronize()

# 使用 vLLM 标准工具比较 logprobs 的接近程度
check_logprobs_close(
    outputs_0_lst=eager_outputs,
    outputs_1_lst=compiled_outputs,
    name_0="eager",
    name_1="compiled",
)
```

评论区精华

gemini-code-assist[bot] 提出两点改进建议：1）compiled 模型和 eager 模型都指定一致的 max_model_len=1024，防止 KV cache 形状差异或 OOM；2）为 eager 模型也添加非空输出断言，避免双方都失败时误判。这些建议在最终提交中已采纳（eager 模型添加了 max_model_len=1024 和输出断言）。

- compiled 与 eager 模型的 max_model_len 一致性 (correctness): 已采纳建议，compiled 模型也添加 max_model_len=1024。
- eager 模型输出有效性断言 (correctness): 已为 eager 模型添加相同断言。

风险与影响

- 风险：风险极低。变更仅涉及单文件测试逻辑，不修改任何生产代码。主要风险是测试覆盖度可能降低：原测试虽然脆弱，但能间接验证模型输出不完全是“垃圾”；新测试只检查 compiled 和 eager 输出在 logprobs 上的相近性，无法检测两者同时出错的情况。但 check_logprobs_close 已经是 vLLM 中广泛使用的测试工具，可靠性有保障。
- 影响：影响范围仅限于 test_dynamic_shapes_compilation 这一个测试，不会影响其他测试或生产功能。修复后该测试在 torch 2.12 上应能稳定通过，减少 CI 噪音。
- 风险标记：测试覆盖调整，仅影响单测试

关联脉络

- PR #41006 [Model][DSV4] Support base model: 同为近期对编译 / 模型测试的修改，但无直接关联。
- PR #40967 [Model] Add MiMo-V2.5 support: 均为涉及编译路径的测试调整，但内容独立。